On Development of Cache Design

DIAO Zhongpu

Zhejiang University

Dec 22, 2020

Design of Cache

To bridge the gap between processor speed and main-memory latency:

- hierarchy: fast hit time
- per-set basis: high hit ratio

hit time and hit ratio tradeoff

Replacement Policy

To store data that will be needed in the near future and discard those unlikely to be used soon

- full-associative: impartical power, latency, and hardware cost demands
- set-associative: victim is identified within the set

minimize conflict miss

Why Improvement Possible

The fully-associative cache with OPT replacement even reduces miss rate more than a set-associative cache of double its size:

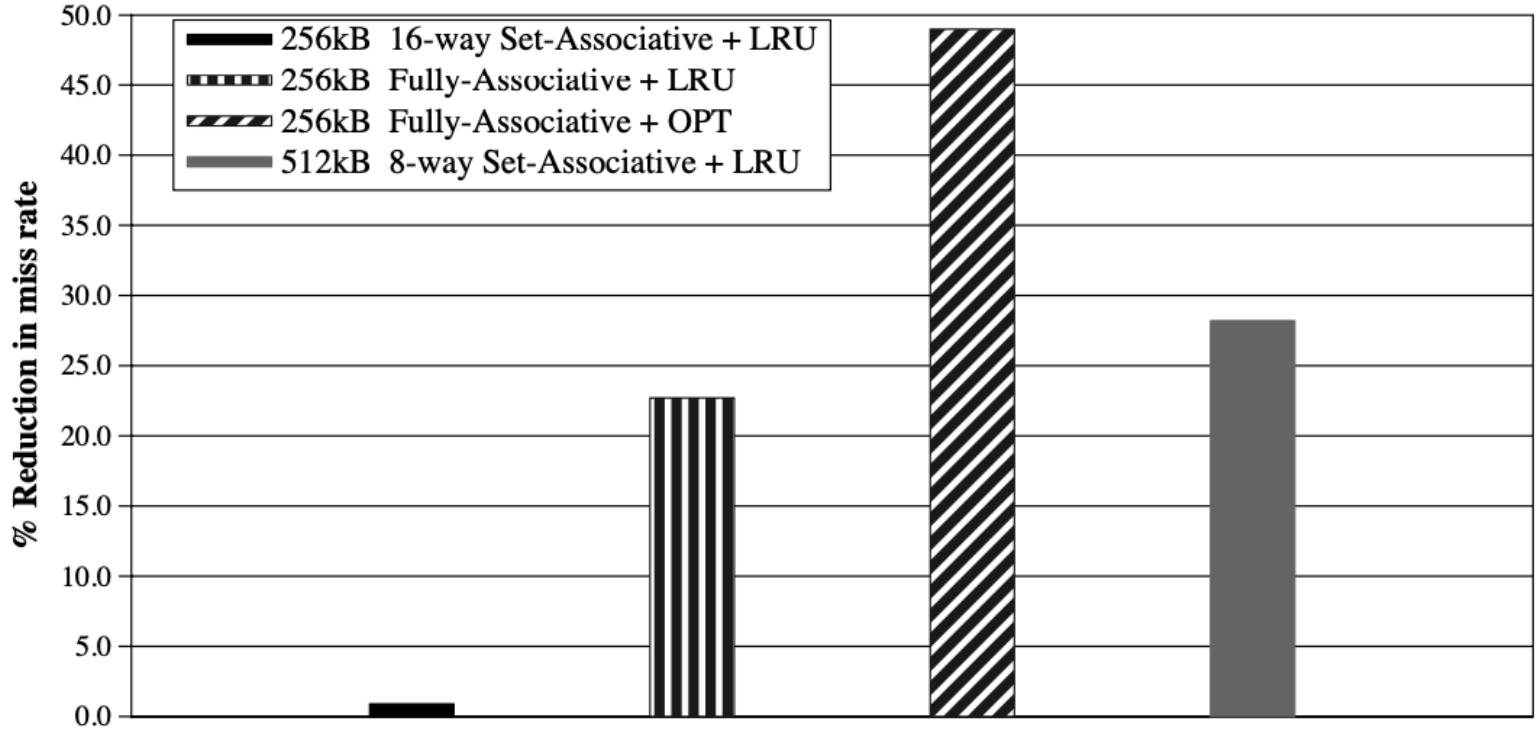
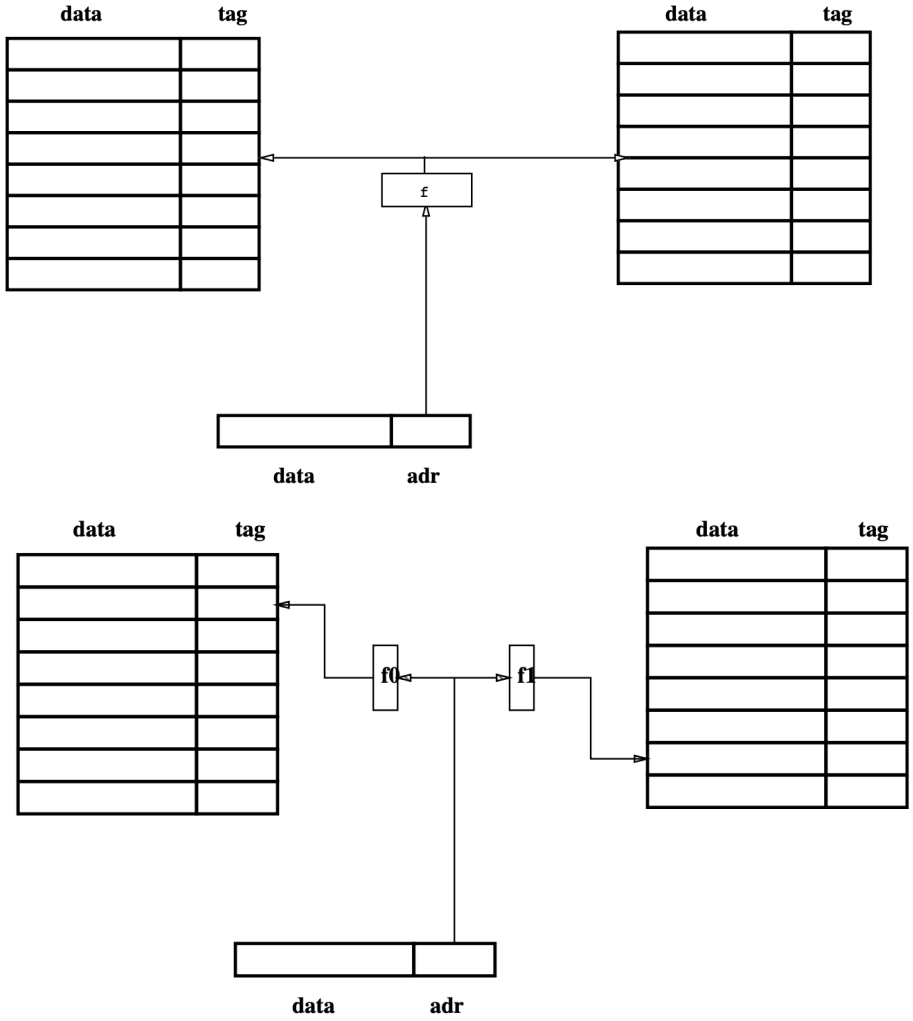


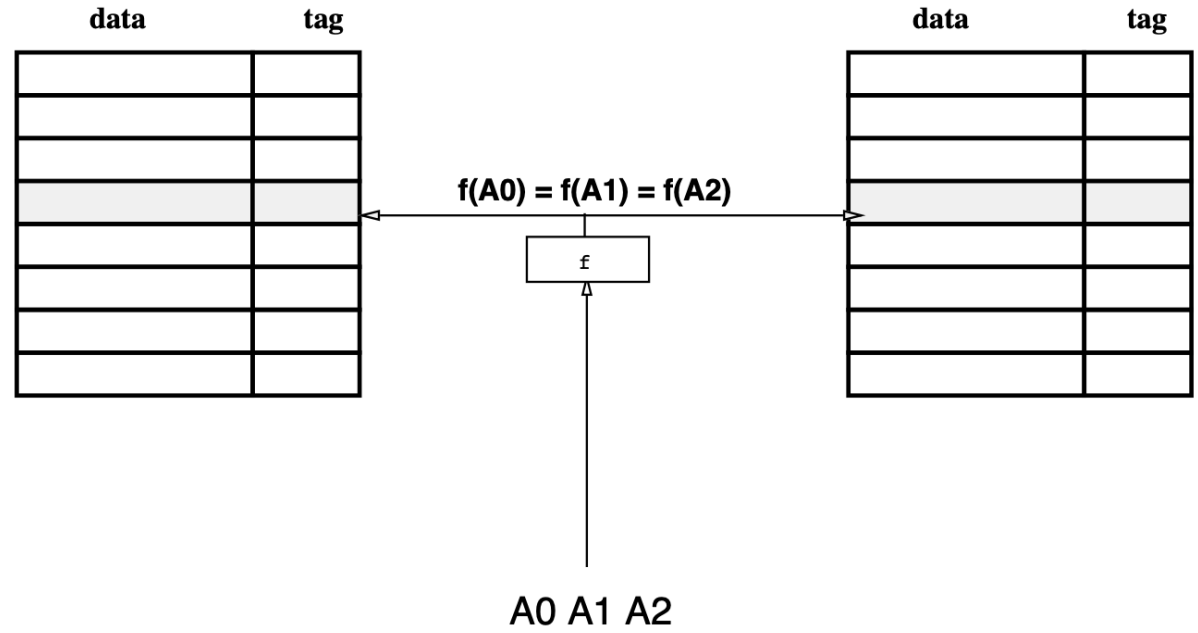
Figure 1. Percent reduction in miss rate compared to a 256kB 8-way set-associative cache.

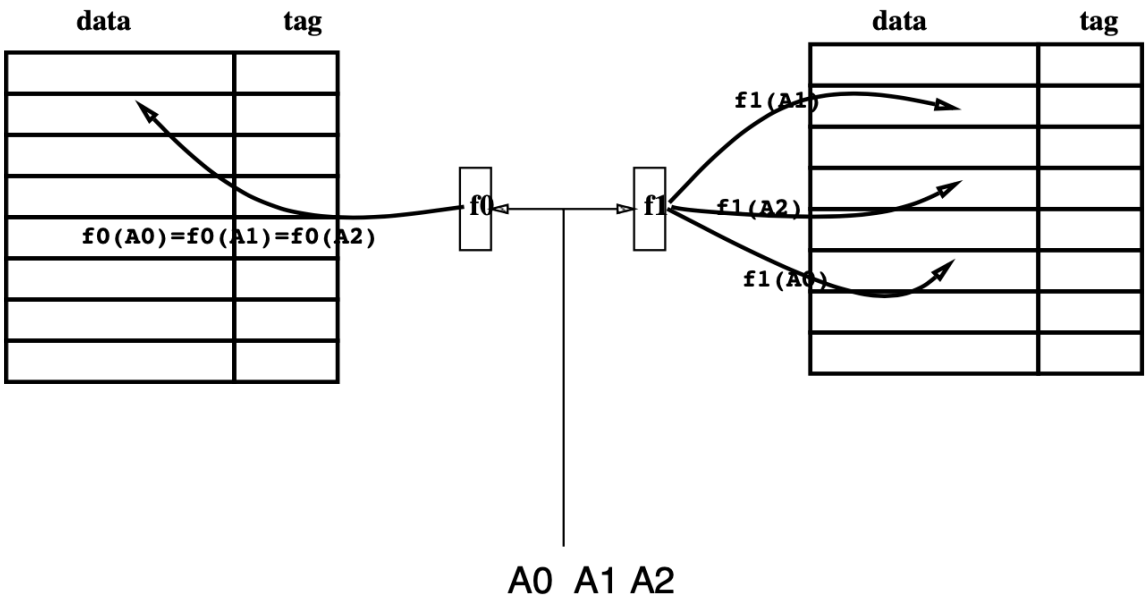
Skewed-Associative Cache

multi-bank cache, mapping lines onto distinct banks

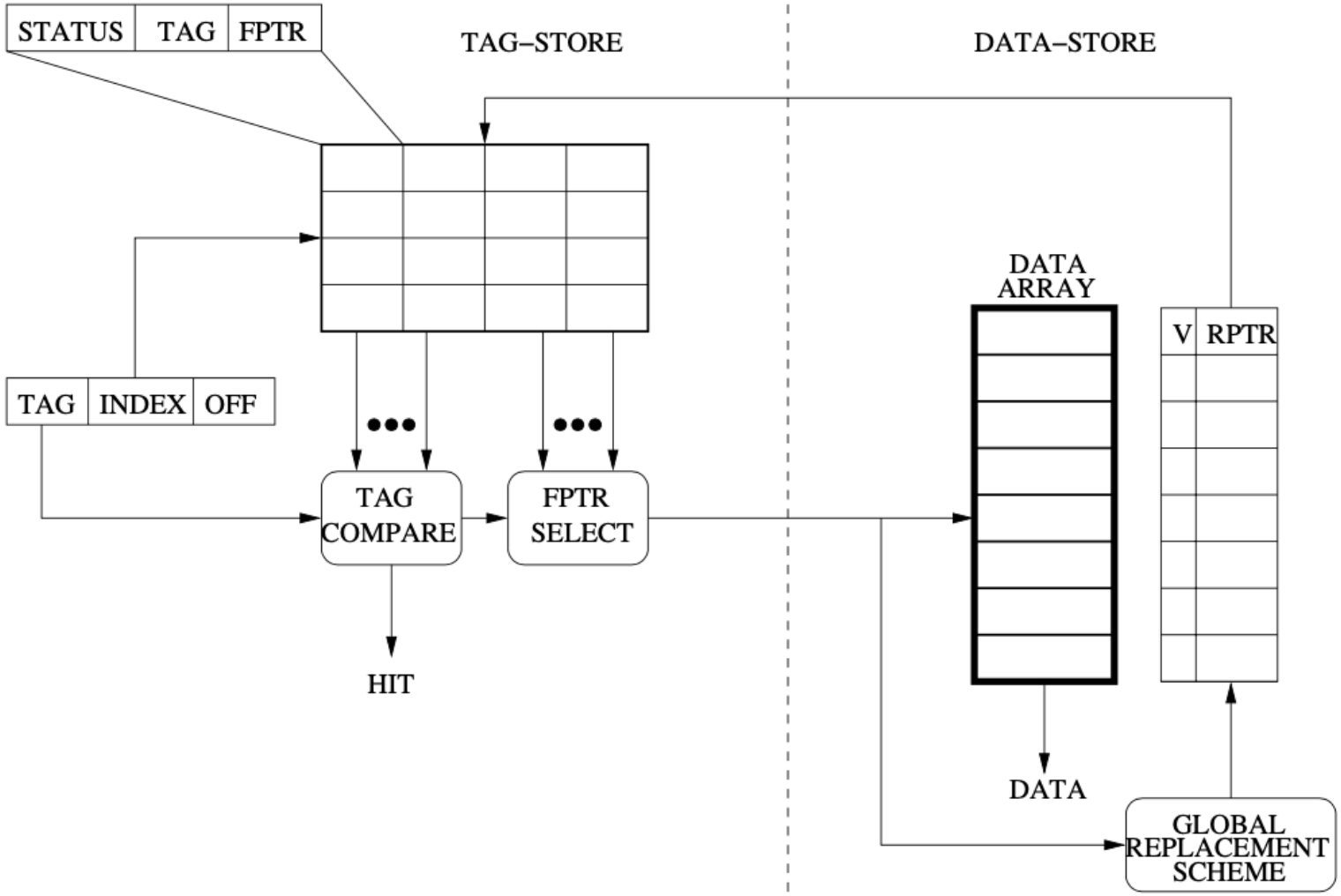


Skewed-Associative Cache





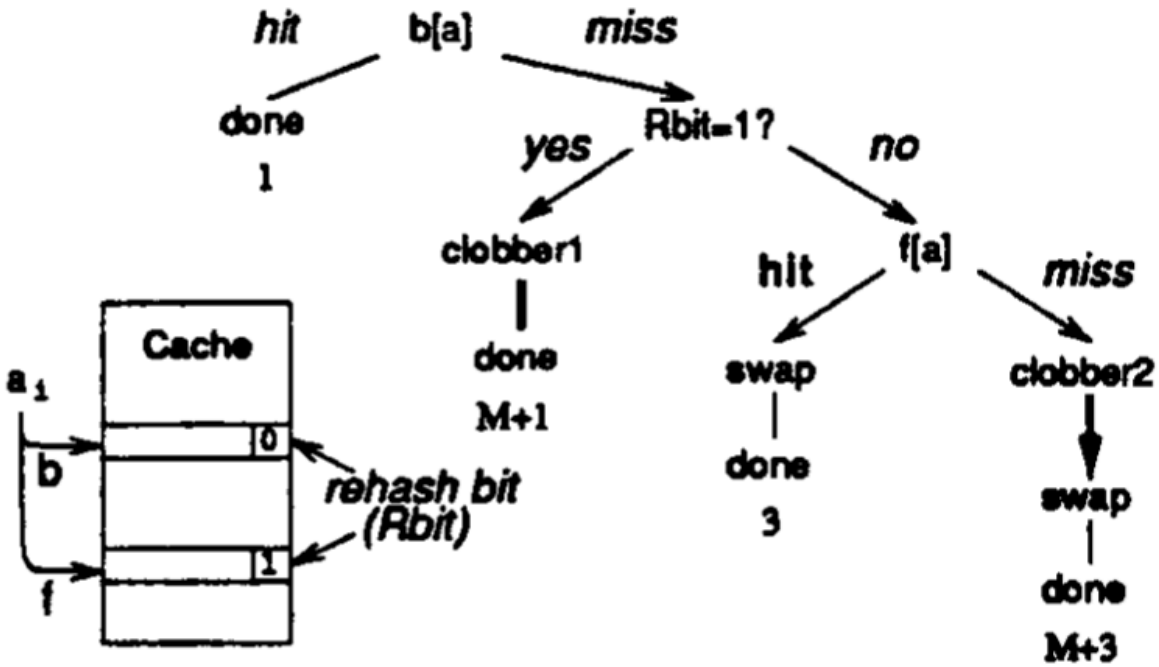
Variable-Way Cache



5.8% storage overhead, 1 extra cycle for cache access (due to tag access, total time 2.45ns->2.64ns for 90nm)

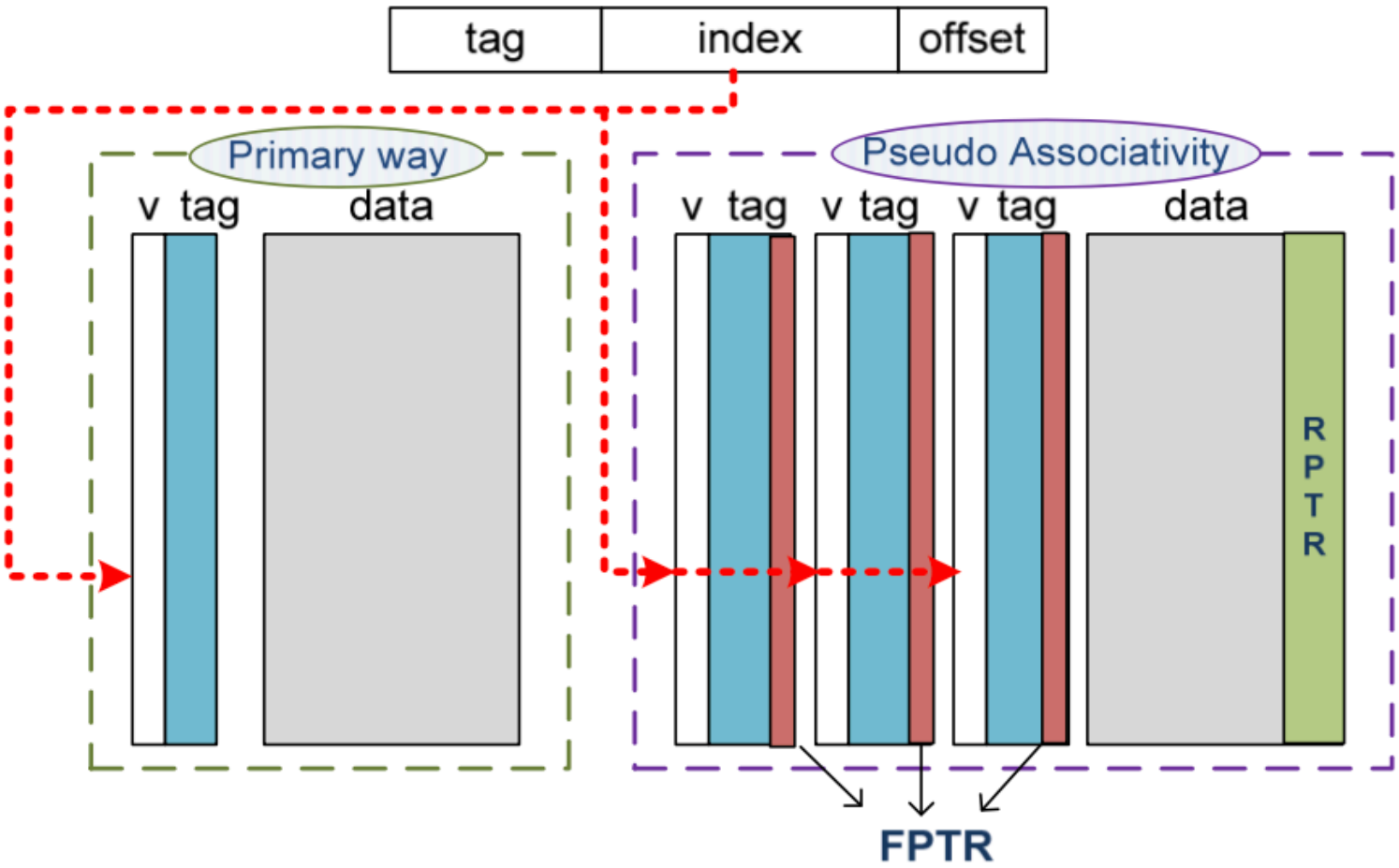
Column-Associative Cache

Column: Possible locations a line can reside in.
Multiple-access cache, with fast access to primary location, pay small penalty if data located in secondary location.



Search Method: rehash bit, index vectors, LRU

(Improved) Pesudo-Associative Cache



fast critical hit time

Variable-Way Cache (also Pesudo-Associative)